



## Las entidades usuario y miigration

### Transcripción

[00:00] ¿Qué tal? Ya vimos cómo es el comportamiento por defecto de Spring Security, vimos que ya nos está generando esa contraseña y por defecto también nos genera una página de login, en caso trabajemos con aplicaciones web, entonces el trabajo ya está hecho pero en nuestro caso que estamos trabajando con API hay un cierto trabajo por hacer y ese trabajo no es como cambiar un valor en el archivo properties.

[00:29] Recuerdan que tenemos es archivo properties y por ejemplo para mostrar el SQL solo cambiamos un flag a true. Esta vez no va a ser así. Requiere una serie de pasos que tenemos que hacer a nivel de código y los vamos a ejecutar uno a uno para que quede muy claro para todos. No es tan sencillo que digamos.

[00:51] Lo primero que tenemos que ver, quiero que recuerden el gráfico que les mostré antes de dónde salen los usuarios en Spring de la base de datos. Se supone que si yo me quiero lograr con mi API, mi API tiene que saber mi usuario y mi clave. Mi nombre de usuario y mi clave para poder comparar y si mi clave es la misma, devolverme mi token.

[01:16] Si yo les digo que necesito eso, ¿qué es lo primero que ustedes pensarían qué debo crear aquí en mi API? Mi migration con mi entidad JPA para mis usuarios. Vamos a crear ahora aquí en mi estructura de mi proyecto, tengo mi dominio y aquí voy a crear un nuevo paquete. Le voy a dar a new package y le voy a poner usuarios.

[01:46] Aquí se me fue un typo. Voy a corregirlo con refactor, rename, y le voy a borrar esta e que no viene al caso, refactorizamos y listo. Y aquí yo le voy a crear una clase, new java class llamado usuarios, una clase normal, aún no la voy a agregar a Git y voy a decirle que tiene atributos private Long id; private String login; que va a ser mi nombre de usuario, private es con minúscula y private String clave.

[02:29] Y todavía tengo mi entidad creada de usuarios. ¿Para que sea una entidad JPA qué más necesito? Necesito las anotaciones de JPA. Entonces voy a entrar A médico y para ahorrar un poco de tiempo lo que voy a hacer es copiar todo esto, vengo aquí y lo pego, pero el nombre de la tabla obviamente va a cambiar. Aquí va a ser “usuarios” y mi entidad va a ser “Usuario”.

[03:00] Aquí hay un error. Yo cometí un error, porque ya el nombre en la clase debería ser en singular, entonces voy a renombrarlo a usuario refactor, listo. Allí tengo mi usuario con mi entidad mapeada, mi tabla. Aquí está correcto @EqualsAndHashCode, generar un getter. Listo.

[03:24] Está todo bien. Y vamos a copiar también los atributos para mi llave primaria de usuarios aquí en el id. Listo, vemos que ya no me da error en usuario porque ya tengo mapeado mi atributo de llave primaria. Esto sería todo lo que tengo que hacer para tener mi entidad usuario ya mapeada en JPA.

[03:49] ¿Qué es lo siguiente que debo hacer? Exacto, el migration. Ahora vengo aquí a los recursos, voy a db.migration y voy a ver que aquí tengo un create-table-medicos, voy a copiar este mismo migration, le voy a pegar aquí, va a ser la versión 4 aquí, porque tengo hasta la versión 3. Yo voy a hacer versión 4: create-table-usuarios.

[04:16] Le voy a dar okay. No lo voy a agregar a Git, ya tengo esto copiado, le voy a decir adiós con id, login y clave, con llave primaria que sea el id, y en el caso de la clave yo le voy a incrementar a 300 caracteres porque yo voy a usar un

algoritmo de Hashing para guardar la clave. No se preocupe, yo se los voy a explicar muy bien luego.

[04:48] No le voy a poner unique porque pueden haber usuarios con la misma clave, entonces no quiero eso por ahora, pero bueno, tengo en login, tengo la clave, not null, otro increment y sería todo lo que yo necesito por ahora. Voy a parar mi aplicación. Creo que ustedes ya saben que cuando trabajamos con migration es bueno detener mi aplicación para ejecutarla.

[05:08] Y eso sería todo lo que tengo que hacer para crear mi entidad usuario. Tengo la entidad de JPA y tengo el migration de usuario. ¿Qué voy a hacer ahora? Voy a ejecutar el migration, Entonces le doy play y vamos a ver qué nos sale aquí en la pantalla.

[05:33] Tengo otra clave generada y vamos a ver. Miren aquí. Satisfactoriamente validadas cuatro migraciones, perfecto, entonces mis migrations fueron ejecutadas. Se aplicó un migration. Ahora está en la versión 4. Parece que me tabla sí fue creado en efecto. Voy a entrar aquí a mi base de datos. Voy a refrescar.

[06:01] Y aquí en mi lista de tablas, esperemos a que refresque, yo debería ver, aquí está mi tabla de usuarios con mi columnitas que yo acabo de crear. Listo. Y eso es todo lo que yo necesitaba hacer como primer paso para transformar mi tipo de autenticación de stateful, stateless. En el siguiente video, vamos a ver un poco más sobre este proceso. Nos vemos.